Responsive web design for Wordpress using Twitter Bootstrap

Robin Klärck

Degree Thesis

Film and Television

2014

Robin Klärck

EXAMENSARBETE	
Arcada	
Utbildningsprogram:	Film & TV
Identifikationsnummer:	4580
Författare:	Robin Klärck
Arbetets namn:	Användning av Bootstrap för responsiv webbdesign med Wordpress.
Handledare (Arcada):	Jutta Törnqvist
Uppdragsgivare:	Documentary & Diversity

Sammandrag:
Responsiv design är ingen ny filosofi inom web design, men aldrig förr har den fått lika hög status som den får idag. Tack vare extensiva ramverk kan man idag planera och utföra sidor i alla storlekar lättare än någonsin förr, vilket för de olika aspekterna av web design varandra allt närmare. I detta arbete har jag tillsammans med Alex Erolin skapat en webbsida för utbytes-programmet Documentary & Diversity, ett projekt med medlemmar från Finland, Ghana och Sydafrika.
Vi kommer att börja med att närma oss projektet från en Mobile First-synvinkel, var vi först bygger sidan med tanke på mobila telefoner samt pekplattor, varefter sidan kommer att utvecklas för bordsdatorer och större skärmar. Efter detta kommer vi att bygga sidan i två stora steg; i den första bygger vi upp en sida som ett test och som en praktisk övning, med hjälp av Unsemantic-ramverket. Denna version kommer att fungera som en inlärnings-fas. I det andra steget bygger vi sidan med Bootstrap-ramverket, vilket kommer att fungera som det slutgiltiga resultatet. I slutet av arbetet går vi igenom skillnaderna mellan ramverken, hur de har påverkat vårt arbetsflöde samt för vilka typer av projekt var sitt ramverk fungerar bäst. Vi bygger sidorna med framförallt en responsiv synvinkel och slutresultatet bör visa denna metod så brett som möjligt.

Nyckelord:	Responsiv design, Documentary & Diversity, Bootstrap, ramverk, online media, web design, front-end development.
Sidantal:	38
Språk:	Engelska
Datum för godkännande:	

DEGREE THESIS	
Arcada	
Degree Programme:	Film and television
Identification number:	4580
Author:	Robin Klärck
Title:	Responsive web design for Wordpress using Bootstrap.
Supervisor (Arcada):	Jutta Törnqvist
Commissioned by:	Documentary & Diversity

Abstract:
Responsive design is no newcomer to the field of web design, however the level of attention and detail it receives today is unprecedented in any form. Thanks to some very extensive frameworks available to web designers today, the design and development of websites of all sizes has become much more practical, and as such has brought the different aspects of web design closer to each other. In this project, me and my partner Alex Erolin have created a web site for the exchange programme Documentary & Diversity, a co-operative program between members form Finland, Ghana and South Africa.

We will start by approaching the project form a Mobile First-perspective, where we will design the site with mobile devices and tablets first in mind, after which we will advance to desktops and larger screens. After this we will build the site in two major steps; the first step will be to build the site using the Unsemantic framework, which will serve us as a learning platform for the second step. Here, we will build the site using Bootstrap, which will also serve as the final draft. At the end of the process, we will compare the two methods and analyze how they affected our workflow, what sort of projects they fit to and the major differences between them. The project will be based on responsive design and the end result must reflect it as well as possible.

Keywords:	Responsive design, Documentary & Diversity, Bootstrap, framework, online media, web design, front-end development.
Number of pages:	38
Language:	English
Date of acceptance:	

INNEHÅLL / CONTENTS

Figures

Tables

1 INTRODUCTION

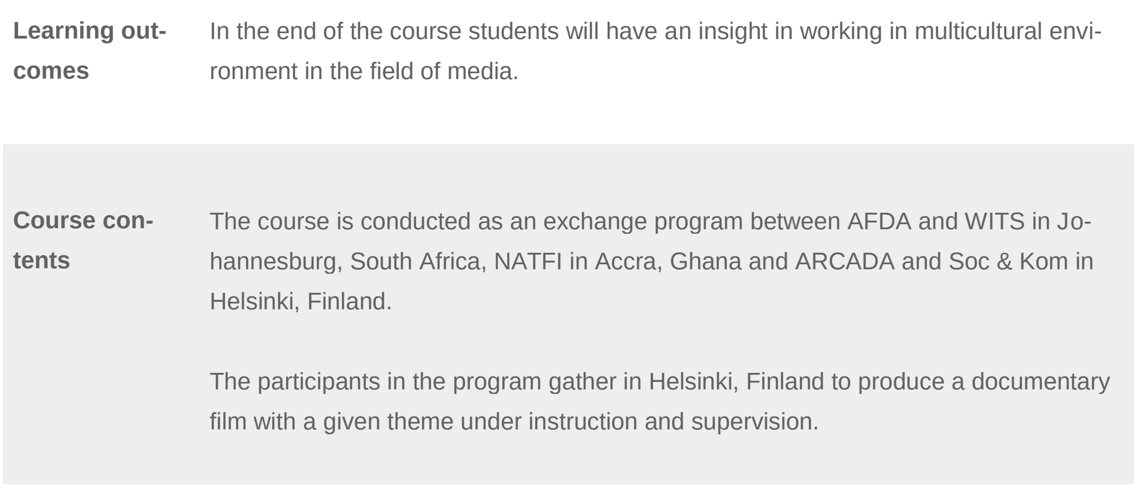
Much like the very extensive frameworks HTML5 Boilerplate or the jQuery library, Bootstrap presents an enormously expansive standard for web development. While the HTML5 Boilerplate concentrates on expanding the possibilities of building a site from the go and the jQuery library offers a vast selection of applications to use for web installations, Bootstrap gives designers a completely new look at flexibility. The concept of Mobile first, the mindset in which web sites should be first designed for mobile phones after which they can be designed for desktops and other platforms, has become the get-go and almost the absolute standard for modern web design and development. As most websites which used to hold separate sites for desktops and mobiles now have either introduced a flexible, responsive site or even going as far as to create an app for mobiles and/or tablets, the separately created mobile versions of websites are dying out. This is a much welcomed change in a field that is constantly evolving.

1.1 Document & Diversity - Project description

Our project part of a programme called Documentaty & Diversity, a co-operative exchange program between several Universities of Applied Sciences which all have Film and Television –based education programmes. These are South African schools AFDA, University of Johannesburg and WITS, Ghanaian NAFTI, and Finnish Arcada as well as Soc & Kom of Helsinki University.

Document & Diversity functions as a course, with the following description:

Competency aims	The aim of the course is to investigate how different cultural environments affects working in production teams with the aim of producing a documentary film.



Learning outcomes	In the end of the course students will have an insight in working in multicultural environment in the field of media.
Course contents	The course is conducted as an exchange program between AFDA and WITS in Johannesburg, South Africa, NATFI in Accra, Ghana and ARCADA and Soc & Kom in Helsinki, Finland. The participants in the program gather in Helsinki, Finland to produce a documentary film with a given theme under instruction and supervision.

Figure 1. Documentary & Diversity course description from the Arcada study guide. 3.4.2014. http://studieguide.arcada.fi/en/curriculumsandcourses/coursedescription/603000/43/TV-3-041/0

In this project, I will create a website which works responsively for all viewports available, and also serves as a social contact hub, with a forum and groups, where the members can communicate and organize movie shoots. They must be able to present their works, pitch their film/documentary ideas, get in contact with people interested in their projects and keep up the process via the hub. This part will be presented via a login function, and as a separate part of the site. The main site will focus on presenting the material which the students upload, the members of the program, current movies under production, archives, FAQ's, contact information and a more thorough look at the program as a whole. The members section will concentrate on functionalities which appear while you're logged in.

1.2 Motivation

On a personal note, a project revolving around the creation of a responsive website which must be designed as a Wordpress installation and then applied for Buddypress compatibility sounds like a large but rewarding challenge. A good web designer, or more specifically said a "Front-end developer", has a very different skillset and know-how today than he might've had just a few years ago. Static websites are seeing a slow decline, and Content Management Systems (CMS) are gaining popularity due to their ease of use and accessibility. Even Graphic designers are generally required to know at least the fundamentals of web design, and be able to code HTML and CSS to some degree. In the same way, a Front End Developer needs much more in his repertoire as well; just HTML and CSS is not enough, knowledge about Javascript, jQuery, even PHP and JSON are gaining a strong foothold in the list of required skills, although the list reaches much further out, sometimes dwelling on the edges of Back-end development.

This project gives me access to a wide set of technology and standards that I had previously not been as proficient at as I might've thought. This also helps explore alternative solutions for certain problems that will undeniably appear in the future.

1.3 Goal

The main goal of this project is to compare two development philosophies in regard of responsive web design and come to a conclusion which version functions better for the specific project. Another main problem to be issued is the use of a free-hand method for creating responsive website against the use of a pre-made web development framework, in this case Bootstrap. I will also explore how to best create this site so that it retains full support in all the modern major web browsers, and that it can be implemented for a Wordpress theme without much trouble.

Therefore, the main problems that I will attempt to solve with this project are:

- How do the popular front-end frameworks **Bootstrap** and **Unsemantic** differ from each other?
- How can these frameworks be utilized to achieve a responsive site?
- Can we bring a Desktop first- layout into a Mobile first- design philosophy?

As this is a client-based project, the results must also present a ready web site with a functioning Wordpress installation as a theme, and adaption for a Buddypress platform installation. Finding out which framework works best for the given task will most likely present a very adapted answer, and as such it may not be perfectly applicable for a large field or other projects. However, for this set project, I may only use one framework in the final draft, and the other version will function as guidelines towards the best version. I will document the process and evolution of both versions, in the end comparing the results and methods with each other. Like this, the result will also yield a comparison between the frameworks within the ramifications of the project.

1.4 Method

As the goal is to create different ways to achieve a responsive Wordpress theme, I will try different methods of achieving this:

The first method will loan from the Unsemantic framework, in which the column and grid sizes are determined by percentages instead of set widths. Like so, it doesn't build on a set amount of grids per column, but instead allows the user to determine the width itself, and bypass the restrictions of a column-based grid system. However, instead of using the framework itself, I will instead create the site using set column widths, in which I will insert grids with percentages as the decisive factors concerning their positions and locations. Like this, I hope to achieve a flexible site where the elements stay in their positions as they should. Moreover, I will give these elements max widths to manipulate their positions on the site as the viewport gets resized. This method is very barebones: I have little previous experience in the subject and as such this method will not be the final draft, and instead functions as a way to get into the system and progress of responsive web design, making clear what the methods are and how they work.

The second method will utilize a grid-based system with a set amount of columns. For this purpose, I have chosen the Bootstrap framework, which works on a 12-column grid system. In this version, I will start the process from scratch, utilizing the different elements offered by Bootstrap to create a similar site as the first one. The versions will be modeled on the basis of the design material offered to me for this project, although with slight modifications, which will be more thoroughly explained in chapter 2.

For the coding purposes, I will use only Notepad ++, abstaining from using other writing software. For image manipulation and design elements I will use Adobe Photoshop and at points, when the software is not accessible, I will use GIMP. For testing purposes, I will use five modern browsers; Google Chrome, Mozilla Firefox, Opera, Apple Safari and Microsoft

Internet Explorer. Tests will be made to make sure the version also functions responsively on the Internet Explorer 8 browser, which is older and retains fewer attributes than its newer counterparts, but still remains one of the most popular web browsers in the world.

Table 1. Web browser usage in early 2014. http://www.w3schools.com/browsers/browsers_stats.asp

2014	Internet Explorer	Firefox	Chrome	Safari	Opera
March	9.7 %	25.6 %	57.5 %	3.9 %	1.8 %
February	9.8 %	26.4 %	56.4 %	4.0 %	1.9 %
January	10.2 %	26.9 %	55.7 %	3.9 %	1.8 %

Once the website is done, it will be ported to Wordpress, where me and my project partner will work on it and implement Buddypress into the theme. Therefore, Wordpress will be used as a Content Management System, and Buddypress will be used as the social communication platform.

1.5 Material

I will mainly use two books for this project to draw guidelines from: Ethan Marcotte's ”Responsive web design” and Luke Wroblewskis ”Mobile first”. Other books that I will use include ”Bootstrap” by Jake Spurlock, ”Wordpress theme development” by Rachel McCollin and Tessa Blakeley Silver, and ”Twitter Bootstrap Web Development” by David Cochran. Since a lot of documentation exists on the official websites of both Bootstrap and Wordpress, I will use much of that code to create the Bootstrap version of the site.

Concerning the main design and visual aspects of the site, these have already been provided, courtesy of Tony Wainio and Malin Kankaanpää, two Arcada students who participated in the Documentary & Diversity-program. The visual material has been approved by the client and works as the principal design for the website, including pages and material.

The text material for the site will be provided by Arcada teacher in Film and Television, Jan Nåls, who is of the main figures behind the Documentary & Diversity program.

1.6 Delimitations

This thesis will only concentrate on the initial part of the project; the process of creating a Wordpress theme out of the ready static site will not be included. The Buddypress installation and application will also not be included. This written part will therefore concentrate on explaining the background work regarding the project, go through the attributes of grid and fluid-grid frameworks which will be used as tools for the project, describe the process which will be followed to create the site and conclude with a comparison of the two models created.

1.7 Terms & Definitions

This thesis assumes that the reader has some basic knowledge regarding web design and development, and understands some of the practices in the creation process of websites. The summaries are based on the descriptions of each definitions individual Wikipedia page/section. The most commonly used abbreviations are:

- **HTML**: Stands for **H**yper**T**ext **M**arkup **L**anguage, functions as the primary coding language in websites. Defines what the site contains and in which order elements appear. Also contains metatext which defines non-visible algorithms and definitions.
- **CSS**: Stands for **C**ascading **S**tyle **S**heet. This file defines how the site looks: from the size of elements to the color of fonts, hovering effects and the like, this alongside HTML-files can create fully functional and stylized websites.
- **jQuery**: A Javascript library with built-in tools for the creation of web applications. Javascript is a syntax-based browser-side coding language for creating animations, commands, functionalities and backups.

- **CMS**: Stands for **C**ontent **M**anagement **S**ystem, functions as an application with which one can easily and quickly modify the content of a website without having to go into the HTML or CSS.
- **Wordpress**: A CMS utilizing the scripting language PHP, Wordpress is one of the most popular CMS's and while its main functionality revolves around blogging, it functions well for smaller and medium-sized webpages. Wordpress utilizes **themes**, which in simpler terms work as a set of clothing for the websites contents. Switching themes may change the visual appearance of the site, but will retain most of the content, such as text, navigation and menus.
- **Buddypress**: A plugin for the Wordpress CMS, it functions as a logon-enabled communication hub, where users can chat, create groups and share content amongst each other.
- **Responsive web design**: As a short summary, responsive web design is the definition of creating web sites which adjust and align themselves according to what media tool you use for viewing it. There are different methods of achieving this goal, and applying it to all elements on a page can pose many problems with modern cross-compatibility between browsers and operating systems. The methods I will explore in this project will concentrate on two main parts.
- **Grid**: The grid-based method works with pixels and pixel definitions. The elements are within a div (wrapper) surrounding or encompassing the grid system, and the columns inside the grid system flow according to the individual column sizes. For example, the most common grid-based systems are 960 pixels wide. Within, they hold 12 columns. These columns can then be divided among the elements one wishes to insert into the site, for example 3 different columns with different sizes. These can be divided so that their combined column size adds up to 12. As the viewport gets smaller, these can then be divided and defined to enlarge themselves or contract themselves, placing them above and below each other, where in a large viewport you could see more elements (smaller columns) per row, and in smaller viewports the rows would contain less elements, meaning that the elements that used to be side by side now lie in new rows below each other.
- **Fluid grid**: Unlike a normal grid system, a fluid grid utilizes the whole web canvas, as such allowing for the creation of web elements which are not con-

fined by a column-based system. In this method, the elements adjust themselves according to the parent element, be it a wrapper, container or the like. Unlike a set amount of columns within a row, the divs in a fluid grid system rely on percentages to float around and adjust themselves as such.

1.8 Bootstrap

As summarized on Bootstraps *About* page (http://getbootstrap.com/about/), Bootstrap was created by two employees at the popular microblog social media site Twitter, Mark Otto and Jacob Thornton. Initially created as a toolkit for containing popular frameworks and applications, which usually collided or conflicted with each other, Bootstrap expanded into the most popular project on GitHub and was released as open source in 2011. As of today, Bootstrap works in all major web browsers, and also functions flexibly in Internet Explorer 8. Older versions do not fully support its attributes.

As explained in *Bootstrap* (Spurlock 2013, Chapter 1), Bootstrap bases itself on a 12-column grid construction. Although it wasn't always about responsive design, since the 2.0 version Bootstrap has become responsive and since then has become the most popular framework for creating Mobile first-design. The 12 column design has been used in web design before Bootstrap, to create symmetrical and eye-pleasing results. This 12-grid system works in four different sets of widths, depending on what resolution the readers device has, in other words what device the reader uses. The class definitions work as this:

class=”col-lg-x”

In this case, the column is defined for large desktop screens, hence the class definition of “lg”. The three other sizes are medium “md” for normal desktop width, small “sm” for tablets and extra small “xs” for mobile phones.

The “x” stands for the column size. As some examples, in case *x = 12*, the column would be full width according to the screen. If *x = 6*, it would cover half the screen from the left. By placing two divs with *class=”col-lg-6”* under each other inside the same row, these would each take up half the row.

The responsive element comes up when the sized classes are combined. For example:

class=”col-lg-3 col-md-4 col-sm-6 col-xs-12”

Like this, the column would take up different amount of space depending on the resolution it is viewed in. In a large screen, it would take up 25% of the width, on a normal desktop it would take up one third of the screen, on a tablet it would take half the screen and on a mobile phone it would be full width. This is how the flexibility comes into shape.

The column sizes in Bootstrap work as follows:

Label	**Layout width**	**Column width**	**Gutter width**
Large display	1200px and up	70px	30px
Default	980px and up	60px	20px
Portrait tablets	768px and above	42px	20px
Phones to tablets	767px and below	Fluid columns, no fixed widths	
Phones	480px and below	Fluid columns, no fixed widths	

Table 2. Bootstrap column width definitions and dimensions. http://getbootstrap.com/

Bootstrap media queries in CSS look like this, from the largest, *col-lg*, to the smallest, *col-xs*:

```css
/* Large desktop */
@media (min-width: 1200px) { ... }

/* Portrait tablet to landscape and desktop */
@media (min-width: 768px) and (max-width: 979px) { ... }

/* Landscape phone to portrait tablet */
@media (max-width: 767px) { ... }

/* Landscape phones and down */
@media (max-width: 480px) { ... }
```

Figure 2. Bootstrap media querie, default definitions. http://getbootstrap.com/css/

Although these widths are changing while the screens of different mediums are becoming sharper and contain more pixels, they can be used as guidelines at this time, and can also be modified in the stylesheets.
The Bootstrap components consist of a LESS-stylesheet, which offers customization for different purposes. Naturally, the grid system limits the possibilities somewhat, although there are very few alternatives for flexible design that stays robust. HTML classes are widely used as they ease the creation of similar objects. Bootstrap classes can be customized with simple adjustments to either the main stylesheets or by creating a custom stylesheet, where one adds definitions to each class.
Bootstrap uses the jQuery library to create flexible and all-browser supported web apps. However, Bootstraps philosophy regarding many of its components is to be as light as possible meaning that instead of using jQuery or Javascript, the components are created in CSS3. This naturally conflicts with many older browsers, meaning that developers have to choose what to prioritize during the development; backwards compatibility or speed.

Grid Template for Bootstrap explains how the Bootstrap grid works:

.col-xs-6 .col-md-4 .col-xs-6 .col-md-4 .col-xs-6 .col-md-4

This is how a row with some content per column would look like on a medium sized desktop. Here the defining class is *col-md-4*. On a mobile phone the same would look like this:

.col-xs-6 .col-md-4 .col-xs-6 .col-md-4

.col-xs-6 .col-md-4

Here the defining class is *col-xs-6*. In this case it would be feasible to change the last divs class from *.col-xs-6* to *.col-xs-12*, which would mean that it covers the whole width of the screen when viewed on a mobile phone. Alternatively one can add a fourth column, which however would mean that the *.col-md-4* class should be changed to reflect that to *.col-md-3*. To avoid problems like this, a Mobile first approach would help lower eventual conflicts and problem situations.

1.9 Unsemantic

Unsemantic is a CSS framework which emphasizes percentages to create the framework. Instead of using columns like Bootstrap does, Unsemantic uses numbers representing percentages, wherein 100% equals the set width of a div. Unsemantic is the successor to the 960 Grid System.

The following is an example of a simple 3-column Unsemantic row, taken from *http://unsemantic.com/*:

```
<div class="grid-container">

   <div class="grid-25">

      I am 25% wide.

   </div>

   <div class="grid-50">

      I am 50% wide.

   </div>

   <div class="grid-25">

      I am 25% wide.

   </div>

</div>
```

In my first design, I will use a system similar to Unsemantic, where percentages determine the widths and positions of each column, div and grid. However, I will not use Unsemantic as a framework, but rather as a guide for constructing the frame into which the site will be built. The site will be built from scratch.

1.10 Other frameworks

1.10.1 960 Grid

The 960 Grid-system was, and still is, one of the most popular Front-end development frameworks around, and although it has lost some foothold to the massive alternatives, it is still standing around strong. Progression towards new mediums within the field is usually slow, and this has benefitted 960.gs. Although very extensive, it does lose some ground to the complete frameworks that are HTML5 Boilerplate and Bootstrap. Thematically, it works in the same way as Unsemantic does, though with less specific code and fewer options to modify.

1.10.2 HTML5 Boilerplate

Along with jQuery library and Bootstrap, the HTML5 Boilerplate has become sort of a go-to tool when a designer starts a project. Boilerplate offers a wide selection of possibilities regarding the initial go, and much like Bootstrap, comes with a set of classes and standards that are easily adaptable to most modern web projects. Recently, Boilerplate has adapted the possibility to use Bootstrap in its ready designs, although its original responsive setup is still available. Boilerplate concentrates less on responsivity and styling, opting instead to emphasize tags and classes, and as such offer the user as plenty of tools as possible to create unique sites, within a certain range.

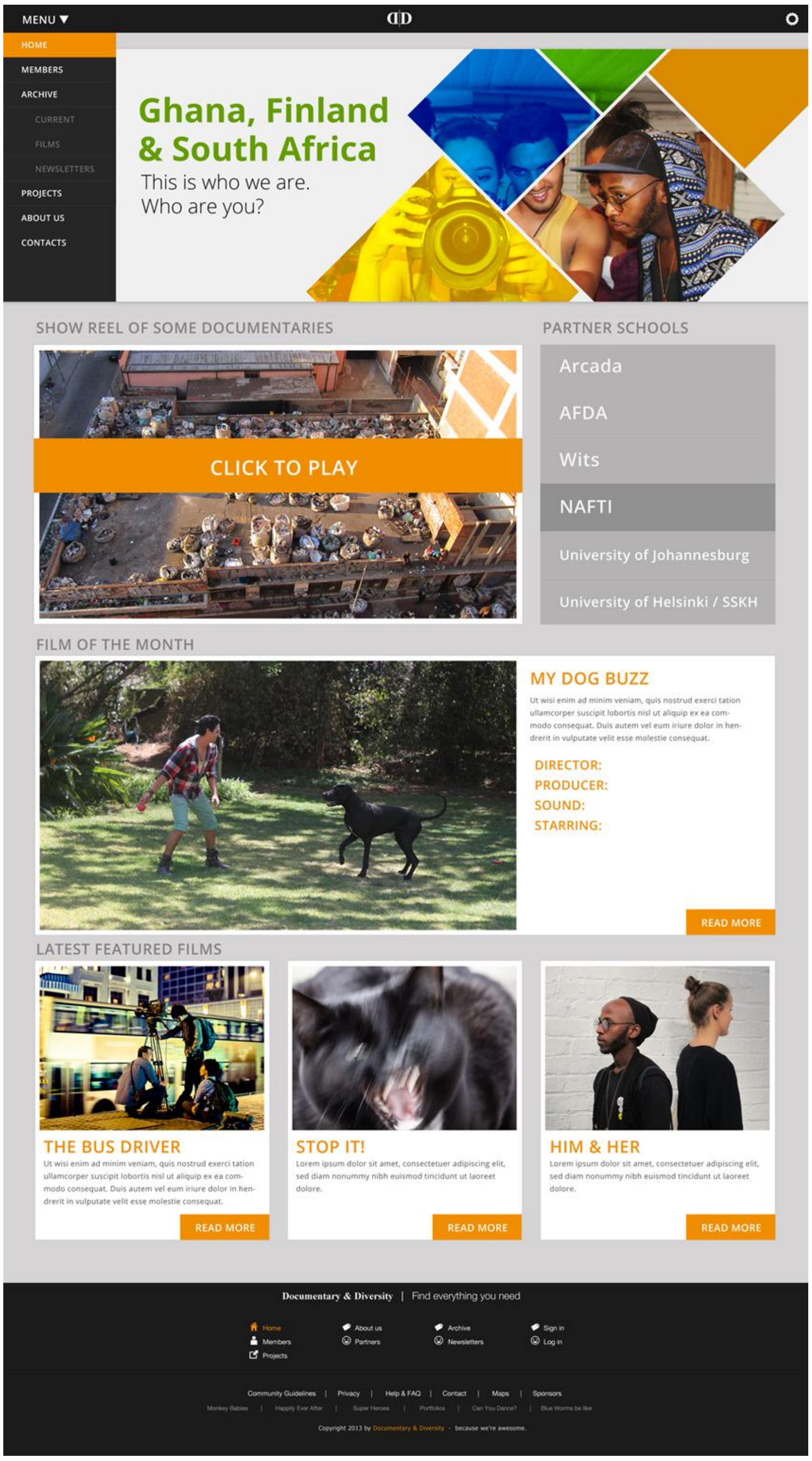
The responsive grid builds on a similar basis as Bootstrap does.

1.10.3 Grid calculator & Gridulator

While neither Grid Calculator nor Gridulator create any CSS or LESS files, they are still included here, as they bring important tools for designers. In short, they work in a similar s 960.gs, offering different sets of grids and widths with their own adjustable offsets for web designers to use in different software, especially Adobe's Photoshop and Illustrator. With these, you can download a .png-file, which then functions as a skeleton for building and designing a website. Like this, the designer can decide how many grids to base the design on, and adjust the grids for different viewports.

These are not frameworks, however they are an integral part of most designers toolbox.

2 PLANNING PHASE



Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore.

Copyright 2013 by Documentary & Diversity - because we're awesome.

Figure 3. Original draft of the index page, created by Malin Kankaanpää, September 2013.

The initial research mostly encompassed the Document & Diversity-program. Neither me nor my partner had attended the program so our first contact with the subject came from our supervisor and students who had attended the program. In short, the Document & Diversity-program is a co-operation between South Africa, Finland and Ghana, where a few film-oriented schools of curriculum educations can send students to work on a more international platform. During the course, students create documentaries and short movies, which generally revolve around their experiences during the visit, and showcase the cultural differences and similarities between the students.

According to *Designing for Emotion* (Aaron Walter, 2011), a site has a few qualities that it needs to make a compelling, attractive layout for a site, especially *Emotional attachment* and *Personality*. Initially, two students had worked on the project before it was taken over by us. The original layouts can be seen at the start of Chapter 2.1.

A mobile version of the site had been planned and lightly developed, and the designs for the site existed as well (see chapter 2.1). Since these designs had been approved by the client, there was no need to further design the site. However, during our preliminary planning, we analyzed the site and its potential flaws. By studying the Mobile first-approach, documented in Luke Wroblewski's *Mobile First*, we drew many contrasts to how the site was initially planned, and how it should've been planned for a Mobile first-approach.

One of the major inconsistences we found was the general UI design of the site and it's interactive elements;

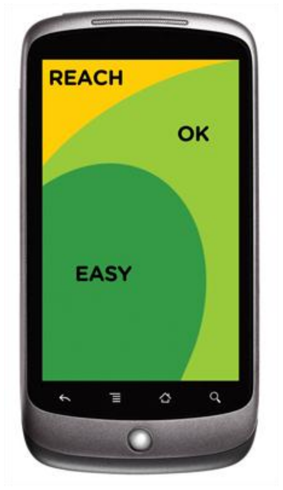


Figure 4. Ease of reaching specific areas of a smartphone with your thumb for right-handed people. Luke Wroblewski, Mobile first, 2011.

According to Wroblewski (Mobile First, 2011, p.72), between 70-90% of smartphone users tend to be right-handed. When using a phone with your right hand, the area in the upper-left corner is the hardest part to reach. In the original design, the dropdown menu was placed in the upper-left corner. Seeing as how users will press the menu button a lot, as it is the easiest way of navigating the site, it poses a problem when it is hard to reach. The login function would be used less, and therefore it could be placed in a less reachable place. Although it strides against good practice, it was a solution with fewer cons than pros, compared to other options we had. The login section was moved to the upper-left corner.

The small change modifies the user experience, especially on mobile phones as the menu is now much more reachable. The login section was moved to the left, as it will not be used as often as the menu dropdown. The change of these elements was also debated against; for left-handed people, this was obviously a drawback, and there was no good way to modify it unless we wanted to move the menu to the middle. However, as this would alter the entire design and we would most likely have to make significant changes to the layout and slides, we decided against it.

The index page differentiates itself the most from the other pages the most, and the other pages are built to be similar to each other, making eventual modifications easier. The page constructions will be explained in chapter 2.1.

List of pages:

- index
- members
- current
- FAQ
- partners & contact
- pitches
- archive

The following are pages visible only in the login section:

- profile
- projects
- ideas
- forums

2.1 Page layouts

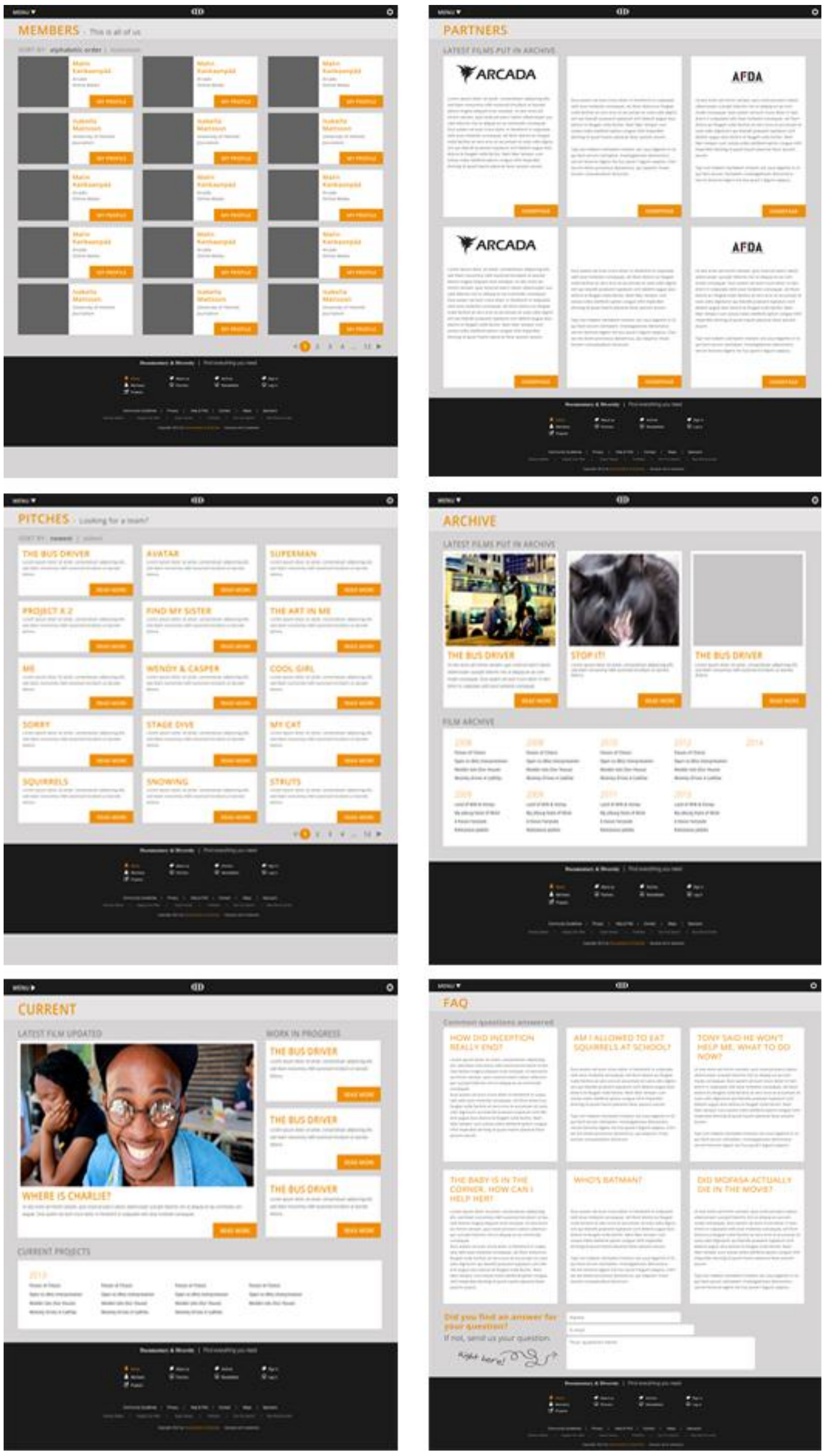

MEMBERS - This is all of us
PARTNERS
LATEST FILMS PUT IN ARCHIVE
ARCADA
AFDA
PITCHES - Looking for a team?
THE BUS DRIVER
AVATAR
SUPERMAN
PROJECT X 2
FIND MY SISTER
THE ART IN ME
ME
WENDY & CASPER
COOL GIRL
SORRY
STAGE DIVE
MY CAT
SQUIRRELS
SNOWING
STRUTS
ARCHIVE
LATEST FILMS PUT IN ARCHIVE
THE BUS DRIVER
STOP IT!
FILM ARCHIVE
CURRENT
LATEST FILM UPDATED
WORK IN PROGRESS
WHERE IS CHARLIE?
CURRENT PROJECTS
FAQ
Common questions answered
HOW DID INCEPTION REALLY END?
AM I ALLOWED TO EAT SQUIRRELS AT SCHOOL?
TONY SAID HE WON'T HELP ME. WHAT TO DO NOW?
THE BABY IS IN THE CORNER. HOW CAN I HELP HER?
WHO'S BATMAN?
DID MOFASA ACTUALLY DIE IN THE MOVIE?
Did you find an answer for your question?
If not, send us your question.
Right here!

Figure 5. Original design for the site, created by Malin Kankaanpää. September 2013.

2.2 Site elements

The site has a few noteworthy elements which can be separated into different parts. Individual columns are not included, as they follow the same formula for most cases.

2.2.1 Navigation

The navigation menu consists of the main pages, including the home page. A logo will be featured, which will be linked to the index page. It follows a gray color scheme. Once minimized in a small viewport, it collapses and produces a clickable navigation, which will open downwards while activated.

2.2.2 Carousel

The carousel will be set to a fixed height, and always at 100% width. It is placed outside the main body content and as such will be a plugin of its own. The set amount of images in the carousel and its slider timings are not specified, and so leave the options open for debate.

2.2.3 Main body columns

The main body columns will be defined using the Bootstrap column classes to create a responsive design. All pages will follow the same formula, although the locations, sizes and content will change according to each page's needs.

2.2.4 Footer

The footer will contain all the pages which exist in the header navigation, but will also contain some pages which concentrate more on administrative subjects and pages that are otherwise necessary but not a priority. The footer will contain eventual links to social media sites and contact information, as well as copyright texts.

2.2.5 Contact form

The contact form will be written in HTML, according to a Bootstrap guideline, stylized in CSS and the back-end will be written in PHP. It will scale according to the viewport and will allow the user to change around some of the field sizes. The initial form was created the *Bootstrap form builder* from *http://bootsnipp.com/forms*, although it was stylized with the guidelines from the official getbootstrap site.

3 FLUID-GRID VERSION

Setting up the site would consist of two main versions, divided into smaller phases. The first main version is not to be used as the final product, but will instead function as a practice round for the actual site construction. Therefore, this version is allowed to fail in a sense, and be less functional than required, as no customized back-end material will be made for it. This will be saved for the main site and the final version. The differences between these two versions will be in the construction code and framework used, as the first version will use no set framework. The second, actual version will utilize Bootstrap to create the flexible responsive design the site requires.

This version can be found at: *http://people.arcada.fi/~klarckro/Slutarbetet/index.html*.

The first version of the site will utilize percentages within percentages to test the flow of the div construction. The site will therefore contract itself according to these set parameters. I have looked at Unsemantic and its philosophy regarding flexible grid-based design, and will create using these methods.

Starting off by creating the background and columns, the site consists of simple divs id's, not using classes. This later proved to be very problematic, as so many elements required similar styling, or in this case had the advantage of requiring similar styles. Therefore the types did not fully work together.

One of the largest problems to solve was the small orange buttons. By default HTML divs start off from the right and on top, when no position styling is applied. The buttons were placed below any images inside the divs, making them appear in places that they were not meant to. We got around the problem by defining their positions with percentages, for example *"{top: 95%;}"* , since the bottom style does not exist. However, this affected every other element as well, meaning that our initial design construction had to either be altered or rebuilt. Seeing as how the first model was doomed to fail, we had to reconstruct it completely for the second model. This proved much easier with the built-in classes of Bootstrap, which allowed for different ways of conquering the problem.

For the slider, we used a pre-made jQuery plugin called Nivo. (http://dev7studios.com/plugins/nivo-slider/). Previously, Nivo had proven problematic once inserted into a pre-made site. Going into the Javascript was not meant to be an issue, as it is meant to be responsive on the get-go. We initially hoped we could use the slider for the second version as well, however the Carousel template (fig. 6) proved to have one of its own which happened to be integrated. As so, the slider became obsolete for the second version, although we used its size as a model for the Bootstrap carousel.
The main elements to make responsive were the boxes with the content. These held the text and images, as well as the buttons within the divs. In short, most of the content was held within these, and according to the initial design the site would be very box-oriented in that sense. We would have to take this into account as we started defining their sizes for the viewports. Since we didn't draw from Bootstrap and its size definitions, we took and compared sizes from a few different sources.

For example, modern iPhones hold the following dimensions:

Element	iPhone 4S (and earlier)	iPhone 5
Window	320 x 480 pts	320 x 568 pts

(including status bar)

Table 3. iPhone dimensions in the viewport. http://www.idev101.com/code/User_Interface/sizes.html

Making it so, the smallest media would be set to be around 320 pixels. Compared to Bootstraps xs-size, this is significantly smaller, and since many phone models have more pixels/inch these days than just a year ago, the 320px size quickly grew redundant, as we found out. We had to squeeze the screen to very small sizes to see results.

The sm-size was taken from the 720px standard. This, however, was also much on the upper side, although closer than the xs-size. The md-size was taken up from what was perceived as a standard desktop display width, and going along with the 960 Grid standard, making it 960 pixels wide. The largest size, lg, was pushed up to 1200px. This was our only match with Bootstraps definitions.

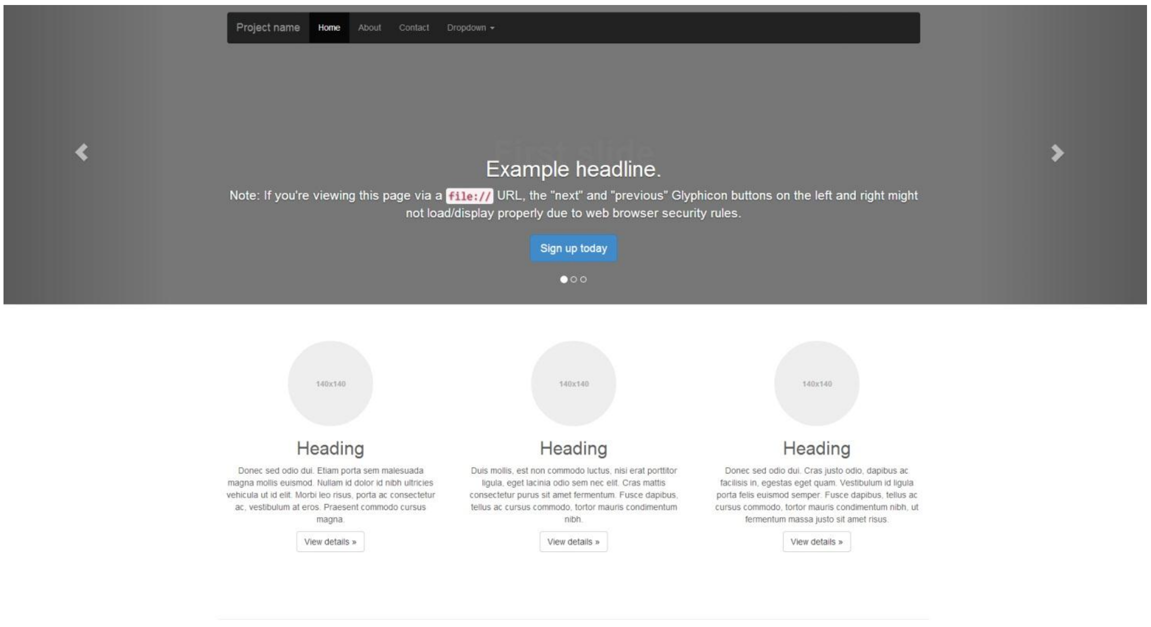
4 GRID VERSION (BOOTSTRAP)

The second (and final) version was created in a few dedicated steps. It can be found at: *http://static.robinklarck.com/*

4.1 Preparation

By searching through models on the official Bootstrap site, we selected one basic template onto which we started building the site. We took into consideration the initial design and all its components, emphasizing certain functionalities like responsivity and the carousel slideshow. We also needed a fixed navigation bar and a large footer into which we would put content. We also had to keep in mind the fact that the site will be transferred over into a Wordpress theme, with Buddypress functionality.

Therefore, a template with a pre-made carousel would be ideal:



Example headline.

Note: If you're viewing this page via a `file://` URL, the "next" and "previous" Glyphicon buttons on the left and right might not load/display properly due to web browser security rules.

Heading

Donec sed odio dui. Etiam porta sem malesuada magna mollis euismod. Nullam id dolor id nibh ultricies vehicula ut id elit. Morbi leo risus, porta ac consectetur ac, vestibulum at eros. Praesent commodo cursus magna.

Heading

Duis mollis, est non commodo luctus, nisi erat porttitor ligula, eget lacinia odio sem nec elit. Cras mattis consectetur purus sit amet fermentum. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh.

Heading

Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

Figure 6. Cropped screenshot of the Bootstrap Carousel template. 4.4.2012.

http://getbootstrap.com/2.3.2/examples/carousel.html

In the first draft, we implemented a slideshow/carousel functionality which was standalone from the internal framework, meaning that it was already included in the package. Since Bootstrap uses the jQuery library and has several options to choose form regarding customization and flexible web app functionality, we decided against using the slider which we had used for the first version, as this could potentially break the site or make it unstable/unresponsive. The template carousel model had to be re-styled and evolved as such.

Although the template had lots of functionalities on its own, most of the elements we needed were not included in the version. Even as Bootstrap offers a sheet with which to create a customized, downloadable package containing all the files for a bootstrap web template, this was not necessary as our plans and approach for the project could possibly alternate during the process. We did not want to fill the site with possible apps which we would not use, so as not to fill up the files and code with needless content.

4.2 Styling the site

Since the bootstrap CSS-file is a large mishmash of code, we had to find an alternative method of modifying the style of our site. On top of it all, as we decided to use the minimized version of the CSS-file, finding elements and classes to modify was hard and uninspiring. A minimized stylesheet is a file with no spaces between the code, which allows the browser to read the file faster. This is a common practice among large sites which are already very heavy, and where every chance to reduce loading times is taken seriously. However, modifying those files is much more troublesome than well styled stylesheets, which usually fit smaller websites better.

To get around this problem, I followed an example explained in *Wordpress Theme Development* (Blakeley Silver & McCollin. 2013. p.150-151). In short, a child theme would be created; by using a blank stylesheet, we could overwrite classes and elements which already existed in the bootstrap.min.css- file, and so modify the elements without going into the Bootstrap code, and potentially damaging the structure of the framework. This was not something I had tried before, however, so unless it worked we had to try to find another way.

Luckily for us, this method worked to an almost perfect degree. Some divs' classes were hardcoded into the bootstrap stylesheet. To get these to work, we had to assign additional classes for a few divs.

4.2.1 Navigation

The template navigation menu is quite different from the one in our initial design. As it is a fixed position, we were worried that moving it too much around would break the responsive element on the bar. In the template version, the menu is already placed in the upper-right corner. As the viewport is reduced, the menu turns into a button, which, when clicked on, pops the menu out below the button, making it functional on mobile phones.

The navigation bar was moved to the top of the screen, and fixed to stay on top. The logo was added to the left by using float, and the listed menu was floated to the right. Background colors were added to each list item to make sure they weren't invisible when the dropdown effect on mobile phones came into play.

4.2.2 Carousel

The carousel was taken straight from the template, as it was already featured in the theme. However, we made several changes to it so that it would not seem copied, and its abilities were not feasible for the way we envisioned the site. The flow and interval between each slide was slightly raised via the corresponding element in the *carousel.js* file:

```
Carousel.DEFAULTS = {
interval: 5000,
pause: 'hover',
wrap: true
  }
```

Previously the carousel slides had a 4 second interval, which we upped to 5. We also changed the transition time of the slides via this element:

```
.emulateTransitionEnd($active.css('transition-duration').slice(0, -1) * 1000)
```

The original one was a bit too long for our vision, so we brought it down to one second.

The carousel was very wide, but its height did not seem proportional. Therefore we had to increase its height, not via percentages but to a set, fixed height. Like so we could import photos into it without having to overstretch them and lose their quality.

4.2.3 Main body

As the site design was grid based, most of the elements in the main body could be easily created with the column classes. On the other pages, this part will be moved around in small doses, mostly just modifying the amount and size of the columns. The index page is a bit livelier, with more images and elements on both sides of the page. The first part below the slider holds an image on the left side, portraying the latest movie to be released. On the right side, all the partner schools are visible. These will stack in a smaller viewport, representing the responsive element of the page. The image was originally planned as a slideshow, however we decided against it, seeing as how the index page would then have one carousel and a slideshow, making it confusing and unnecessary. The image is wrapped within an image class called *.thumbnail*, which creates a white border around it and creates a small gutter around the image. The partner list stands on the side, with a *col-md-3* class definition.

Below are two divs similar to the ones above, just changing around. The larger one stays on the right side and the smaller one on the left.

4.2.4 Footer

The footer proved rather problematic at first; since most of the site was bound within a container which held the width at 90%, the footer could obviously not be inserted into this container. To create a fluid complete package, we searched through Bootstraps functionalities in order to find out how to create a footer that would match the original mockup.

A footer template called “sticky footer” exists, however it proved problematic and much harder to customize than most of the other elements in Bootstrap. We went around the problem by opting not use a premade footer from the framework, but instead constructed our own, utilizing just a few classes from Bootstrap.

5 DISCUSSION & RESULTS

The Wordpress installation can be found here: *http://docdiv.robinklarck.com/*

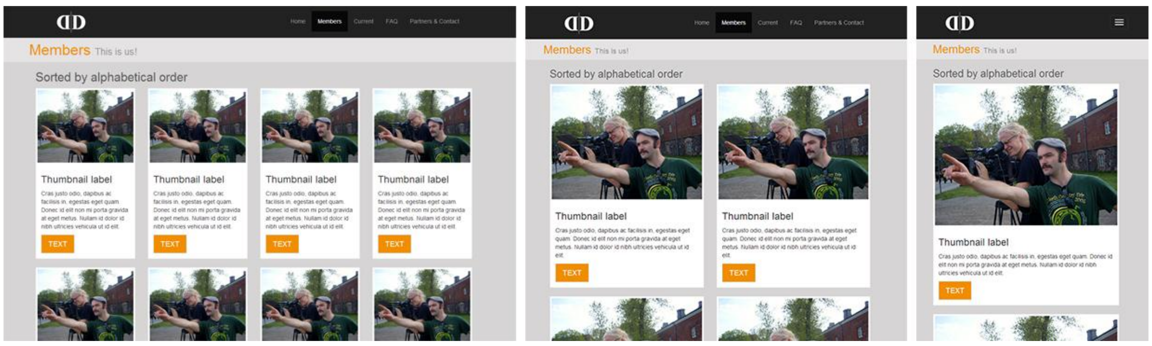


Figure 7. Cropped screenshots of how the responsive elements work. From left to right: medium viewport (980px and up), small viewport for tablets (768px-480px) and mobile phone viewport (480px and below).

In this project I have compared two methods of creating flexible, responsive front-end development for a website. As the main focus lied on creating one site without a steady and stable front-end framework and another with a large, fluid framework, the comparisons will ultimately shift in the latter's favor. As the design schematics remained the same for both versions, no bias could be applied to either version at the expense of the other.

5.1 Fluid-grid method

The results & observations for this version can be separated into two categories; those before the creation of the second version, and those after. Regarding the observations before the second version was created, the non-framework version gave insight into how a grid-based system works. However, seeing as how the Unsemantic framework works with percentages rather than fixed grids, the grid planning was focused more in the layout design instead of the actual coding phase. This caused problems early on and also caused modification of the code by another person to become confusing.

The second observation category revealed many points of improvement, and portrayed the ease of modifying and creating grid-based websites with a fixed amount of grids to work with. The percentage system did retain some advantages, as the grid-based framework does not allow the site to overlap the ramifications of the grid. This is a much smaller obstacle for a percentage-based design, and opens up options when creating sites which go beyond grids.

5.2 Grid method (Bootstrap)

What stood out the most with the Bootstrap version was the ease of creating the grid-based design in HTML and CSS. Very little outer elements were required, and most of the modifications into the classes could be made with a custom stylesheet, placed on top of the other ones. Whether or not this can be considered bad practice is up for the developer to choose, seeing as how it barely affected the loading times of the site and allowed the original bootstrap.css -file to function as a backup in case of too many modifications. Therefore, our conclusion is that for a site this size its advantages outweigh the disadvantages.

5.3 Conclusion

While both versions work to create a responsive site, the Bootstrap framework offers the easiest and most encompassing way of creating a responsive grid-based static website using front-end coding. However, when it comes to percentage-based building without a column- and grid- based design scheme, it is highly recommended that the developer uses another framework. For this end, Unsemantic would prove an easier choice, and would most likely offer better tools for the end means. While neither system stands as the end-all toolbox, both offer flexibility in their customization and work well with much of the jQuery library, and also offer some plugins created specifically for each framework. These answers the questions: “How do the popular front-end frameworks Bootstrap and Unsemantic differ from each other?” and “How can these frameworks be utilized to achieve a responsive site?”.

In the end, the approach depends on the planning; as we created the site with responsive design in mind, but also Wordpress and Buddypress, we had to strap down some design elements which could not be implemented within the full picture. A Bootstrap-theme for Wordpress already exists, and we also experimented with it to see how well it works with the material we have received. This may also be used for those who wish to create

a Bootstrap- based Wordpress- site, although the customization potential would be significantly smaller.
As the site was originally designed for a desktop view, it proved to be a bit problematic in adapting it to a Mobile First-approach. However, since most of the elements within the site were squares and in forms which could easily flex themselves within the viewport, it became a process of patience rather than attrition. The site design could be brought over to a Mobile First- approach, as the placement of the elements could be stacked rather fluidly. This approach helped us in the planning and development of the responsive guidelines, and followed well into the higher and larger viewports.

REFERENCES

Web sources

Smith, Nathan. *Unsemantic CSS Framework*. Available at: http://unsemantic.com/

[Accessed March 01, 2014].

Smith, Nathan. *The 960 Grid System*. Available at: http://960.gs/

[Accessed March 01, 2014].

Coyier, Chris. *Don't Overthink it Grids*. Published August 14th, 2012. Available at: http://css-tricks.com/dont-overthink-it-grids/

[Accessed March 03, 2014].

Stuntbox LLC. *Say Hello to Gridulator*. Available at: http://stuntbox.com/blog/2010/09/say-hello-to-gridulator/

[Accessed February 27, 2014].

Nielsen, Nicolaj Kirkgaard. *Grid Calculator*. Available at: http://gridcalculator.dk/.

[Accessed February 27, 2014].

Bootstrap grid examples. Available at: http://getbootstrap.com/examples/grid/

[Accessed February 27, 2014].

Surguy, Maks. Bootsnipp.com. *Form Builder for Bootstrap*. Available at: http://bootsnipp.com/forms/

[Accessed February 22, 2014].

Bynens, Mathias. Maris, Cãtãlin. Reinl, Hans Christian. *HTML5 Boilerplate*. Available at: http://html5boilerplate.com/

[Accessed February 20, 2014].

Literature

Blakeley Silver, Tessa & McCollin, Rachel. 2013. *Wordpress Theme Devlopment - Beginner's Guide.* Packt Publishing, 3rd New Edition. ISBN: 978-1849514224. 252 p.

Cochran, David. 2012. *Twitter Bootstrap Web Development How-To*. A Book Apart. ISBN: 978-1-8495188-2-6. 68 p.

Marcotte, Ethan. 2011. *Responsive Web Design*. A Book Apart. ISBN: 978-0-9844425-7-7. 143 p.

Spurlock, Jake. 2013. *Bootstrap*. O'Reilly Media. Print ISBN: 978-1-4493-4391-0 | Ebook ISBN: 978-1-4493-4390-3. 128 p.

Walter, Aaron. 2011. *Designing for Emotion*. A Book Apart. ISBN: 978-1-937557-00-3. 98 p.

Wroblewski, Luke. 2011. *Mobile First*. A Book Apart. ISBN: 978-1-937557-02-7. 123 p.